**IN THE CLAIMS**

Please amend claims 11 and 34 through 37, as follows:

1. (Canceled)

2. (Canceled)

3. (Canceled)

4. (Canceled)

5. (Canceled)

6. (Canceled)

7. (Canceled)

8. (Previously Presented) The method as recited in claim 11, wherein the merging a result comprises:

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding contents of the first register with the modified second register; and

placing the result in the result register.

9. (Original) The method as recited in claim 8, the method further comprising:

ignoring a carryover of the result from the low-order bit positions of the result register to the high-order bit positions of the result register.

10. (Previously Presented) The method as recited in claim 11, wherein the merging a result comprises:

modifying contents of the first register by placing data values of zero in the low-order bit positions of the first register;

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding the modified first register with the modified second register; and

placing the result in the result register.

11. (Currently Amended) A method of sub-register data operations in executing an instruction, the method comprising:

executing the instruction on a first register and a second register;

disabling a carryover of a result of the executed instruction from low-order bit positions of a result register to [[the]] high-order bit positions of the result register; and

merging the result of the executed instruction with a plurality of high-order bits from the first register, ~~the plurality of high-order bits being copied into high-order bits from the first register,~~ the plurality of high-order bits being copied into the high-order bit positions of [[a]]the result register, and the result being placed into the low-order bit positions of the result register.

12. (Previously Presented) The method of claim 11, wherein the first register and the second register have 32 bits.

13. (Previously Presented) The method of claim 11, wherein the result register has 32 bits.

14. (Previously Presented) The method of claim 11, the method further comprising:

using a renamer to assign the first register, the second register, and the result register.

15. (Previously Presented) The method of claim 11, wherein the result of the executed instruction is less than 32 bits.

16. (Original) The method of claim 15, wherein the result of the executed instruction is less than or equal to 16 bits.

17. (Original) The method of claim 16, wherein the result of the executed instruction is less than or equal to 8 bits.

18. (Previously Presented) The method of claim 11, wherein the merging a result is performed before instruction execution is complete.

19. (Previously Presented) A processor comprising:

an instruction set having an instruction;

a source register and a destination register referenced by the instruction from the instruction set; and

a logic circuit to examine the instruction before execution to identify a portion of the source register that should remain unchanged into the destination register, and the logic circuit further to move the unchanged portion into the destination register before instruction execution is complete, the logic circuit including a carryover circuit to disable a carryover from the execution of the instruction to the unchanged portion of the destination register.

20. (Previously Presented) The processor of claim 19, wherein the logic circuit is to move the unchanged portion into the destination register by setting corresponding values of the source register to zero.

21. (Original) The processor of claim 19, wherein the source register and the destination register have a greater bit-length than a result of the instruction.

22. (Original) The processor of claim 21, wherein the source register and the destination register have 32 bits.

23. (Original) The processor of claim 21, wherein the result of the instruction has less than 32 bits.

24. (Original) The processor of claim 23, wherein the result of the instruction is less than or equal to 16 bits.

25. (Canceled)

26. (Canceled)

27. (Canceled)

28. (Previously Presented) A method comprising:

receiving an instruction to perform an operation on contents of first and second source registers, the contents including a plurality of bits and the operation results being a different bit length than bit lengths of the first and second source registers;

identifying high order bits of one of the source registers that are to remain unchanged when merged into the destination register;

modifying the contents of the other source register by setting corresponding high order bits of the other source register to zero;

adding the contents of the one of the source registers with the modified contents of the other source register;

placing results of the addition in the destination register; and

disabling a carryover of the addition results from low-order bit positions of the

destination register to high-order bit positions of the destination register.

29. (Previously Presented) The method of claim 28, wherein screening the first and second source registers comprises:

modifying contents of one of the source register by setting low order bits of the one of the source registers to zero; and

modifying the contents of the other source register by setting high order bits of the other source register to zero.

30. (Previously Presented) The method of claim 29, wherein merging the operation results comprises:

adding the modified contents of the one of the source registers with the modified contents of the other source; and

placing results of the addition into the destination register.

31. (Original) A machine-readable medium having stored thereon a plurality of executable instructions for performing a method comprising:

receiving an instruction to perform an operation on contents of first and second source registers, the contents including a plurality of bits and the operation results being a different bit length than bit lengths of the first and second source registers;

identifying high order bits of one of the source registers that are to remain unchanged when merged into the destination register;

modifying the contents of the other source register by setting corresponding high order bits of the other source register to zero;

adding the contents of the one of the source registers with the modified contents of the other source register;

placing results of the addition in the destination register; and

disabling a carryover of the addition results from low-order bit positions of the destination register to high-order bit positions of the destination register.

32. (Original) The machine-readable medium of claim 31, wherein screening the first and second source registers comprises:

modifying contents of one of the source register by setting low order bits of the one of the source registers to zero; and

modifying the contents of the other source register by setting high order bits of the other source register to zero.

33. (Original) The machine-readable medium of claim 32, wherein merging the operation results comprises:

adding the modified contents of the one of the source registers with the modified contents of the other source; and

placing results of the addition into the destination register.

34. (Currently Amended) A machine-readable medium having stored thereon a plurality of executable instructions for performing a method comprising:

executing the instruction on a first register and a second register;

disabling a carryover of a result of the executed instruction from low-order bit positions of result register to [[the]] high-order bit positions of the result register; and

merging the result of the executed instruction with a plurality of high-order bits from the first register, ~~the plurality of high-order bits being copied into high-order bits from the first register,~~ the plurality of high-order bits being copied into the high-order bit positions of [[a]]the result register, and the result being placed into the low-order bit positions of the result register.

35. (Currently Amended) The machine-readable medium of claim 34, wherein the merging [[a]]the result comprises:

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding the contents of the first register with the modified second register; and

placing the result in the result register.

36. (Currently Amended) The machine-readable medium of claim 35, the ~~machine-readable medium~~ method further comprising:

ignoring a carryover of the result from the low-order bit positions of the result register to the high-order bit positions of the result register.

37. (Currently Amended) The machine readable medium of claim 36, wherein the merging [[a]]the results comprises:

modifying the contents of the first register by placing data values of zero in the low-order bit positions of the first register;

modifying contents of the second register by placing data values of zero in the high-order bit positions of the second register;

adding the modified first register with the modified second register; and

placing the result in the result register.